Agile Projects
and
Big Data Products

How Agile is relevant

Eileen O’Callaghan

Greeting

Eileen O'Callaghan

Academic - Monash lecturer and tutor for 3+ years

- Professional Communication
- Algorithmic Problem Solving
- Analysis and Design
- Project Management
- Java Programming

IT Manager: NZX Melbourne Office

- Agile Scrum environment

Software Engineer: Telco domain

- Major clients: Telstra, Optus, Virgin, Siemens
- R&D Start Up project: Software Defined Radio

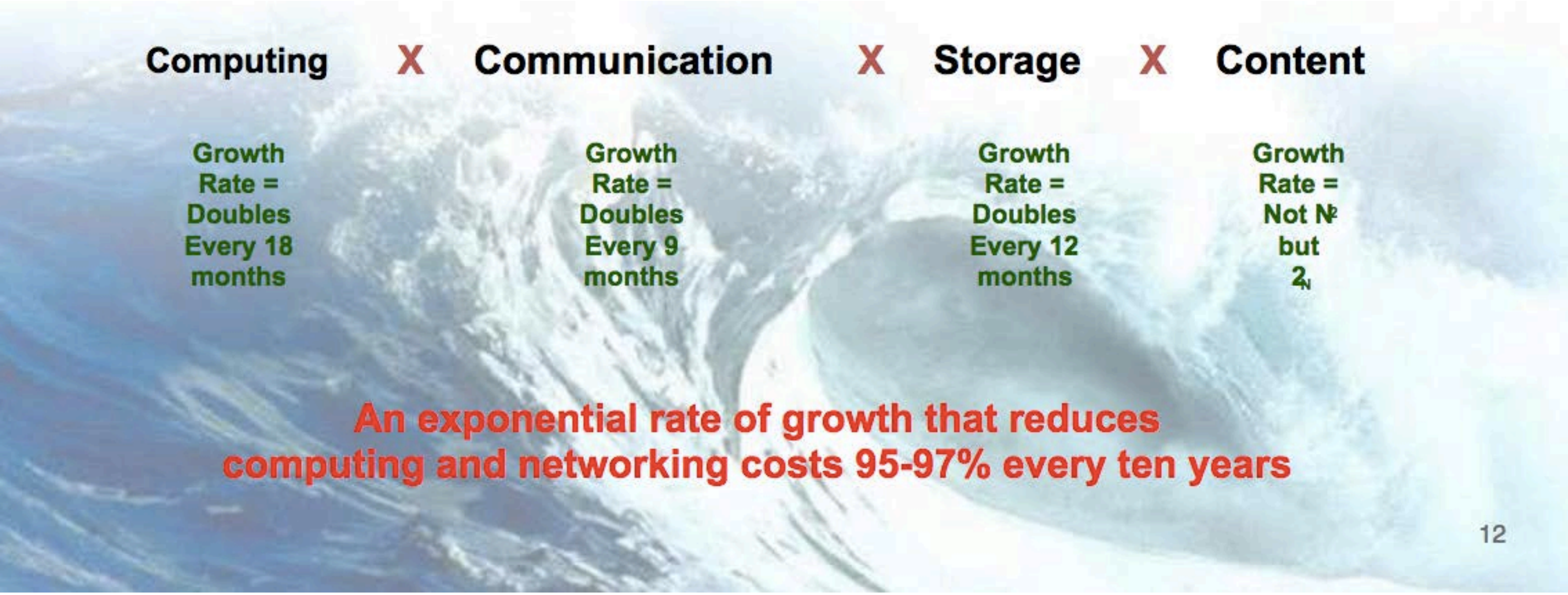
Purpose of lecture

1. What is trending in IT, the knowledge industry?
2. What is happening to products?
3. Why do products?

4. When is an Agile project a good choice?

What: Digital Power

Digital Power is surging forward



Computing X **Communication** X **Storage** X **Content**

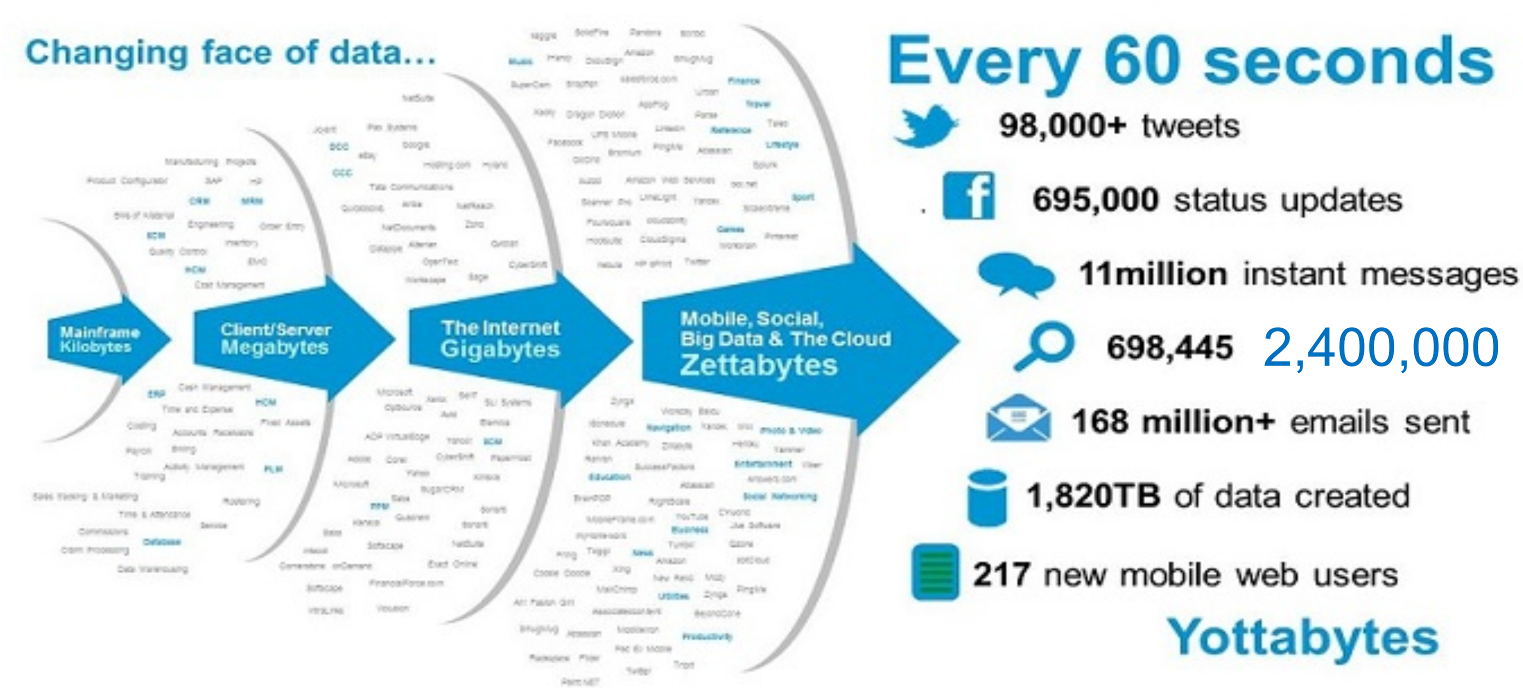
Computing	Communication	Storage	Content
Growth Rate = Doubles Every 18 months	Growth Rate = Doubles Every 9 months	Growth Rate = Doubles Every 12 months	Growth Rate = Not N^2 but 2_N

An exponential rate of growth that reduces computing and networking costs 95-97% every ten years

12

Big Data Ecosystem

- Data growth is exponential
- Data is structured, unstructured, geospatial
- Available from fast evolving Internet technology



2013

2,400,000 2017

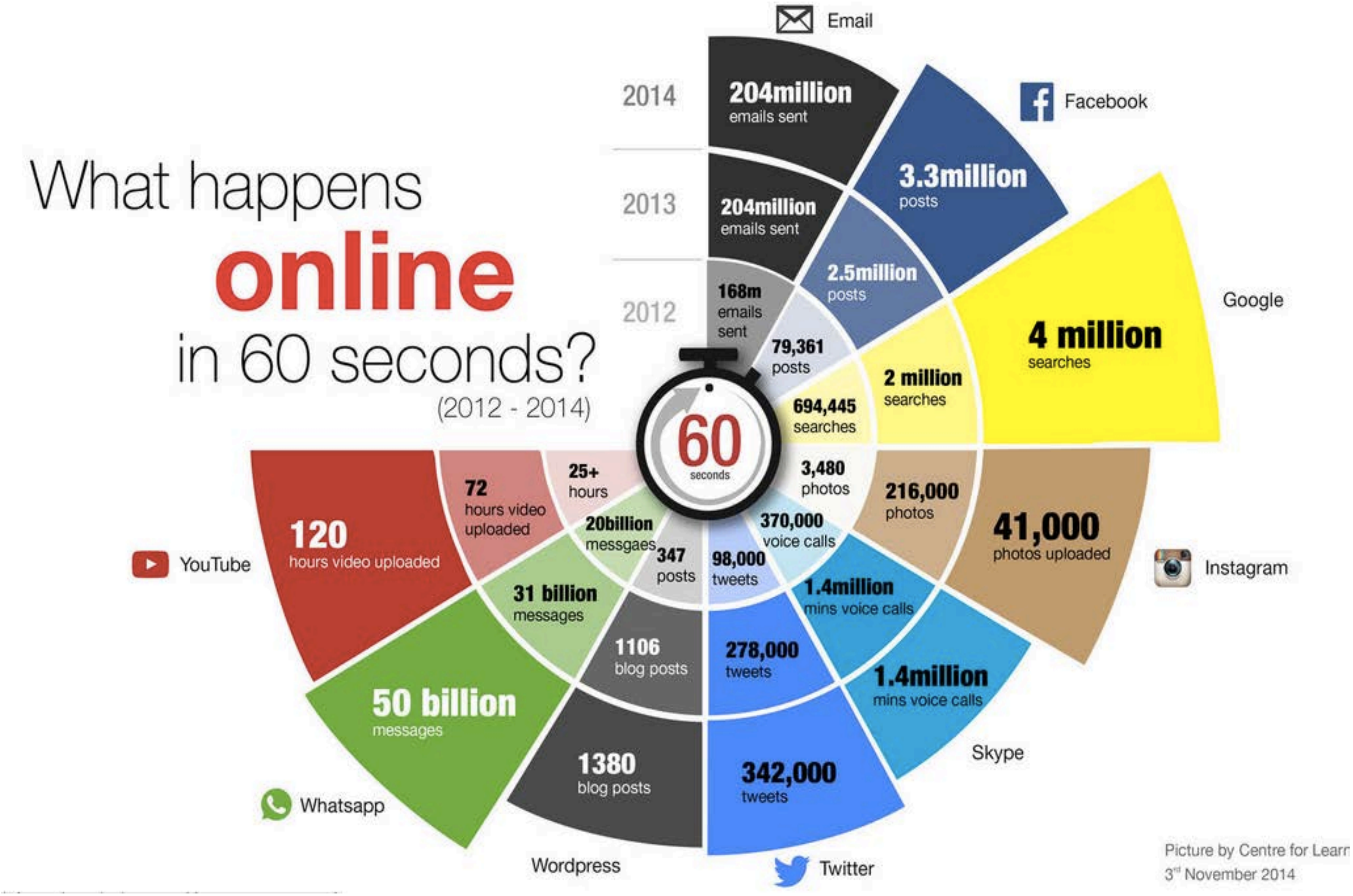
Kilo → Mega → Giga → Zetta → Yotta

http://www.irlogi.ie/wp-content/uploads/2013/10/Big-Data-Analytics-and-GI-KieranTowey.pdf, October, 2013

Big Data Technical Challenge

- Store volume
  - global cloud infrastructure is cheap and big
- Integrate a variety of messy, scattered data
  - deep interoperability of data types
- Manage velocity
  - near real time analytics avoids stale data
  - visualization tools allow humans to interpret data

Big Data Visual Interpretation



Picture by Centre for Learr
3rd November 2014

Big Data Value Challenge

- Data becomes stale and irrelevant quickly
  - seek volatile trends in real time
  - maximize value of transient opportunity
- Low density information, redundant information
  - automated pattern recognition
  - data enabled strategic decisions and predictions
    - fewer gut reaction decisions
- Know your potential customer
  - narrow target using accurate details

What: is happening to Products?

Identify specific consumer groups

- Target with novel products
  - Move with care and great tact
- Exploit transient trends
  - Consumer herd behavior
  - Big Data orientated new products

"viral" marketing

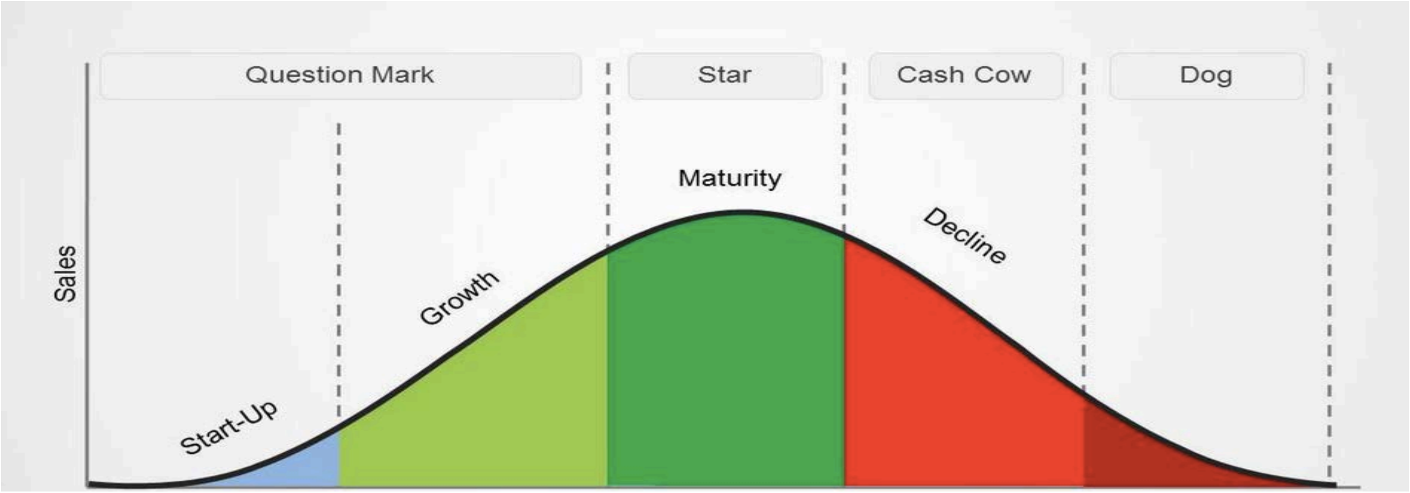
- Constantly respond and adapt product
  - Competitive advantage to identify new market early

New Opportunities!!

Products for transient opportunities

Shorter interval product lifecycle

- Re-purpose quickly => PIVOT

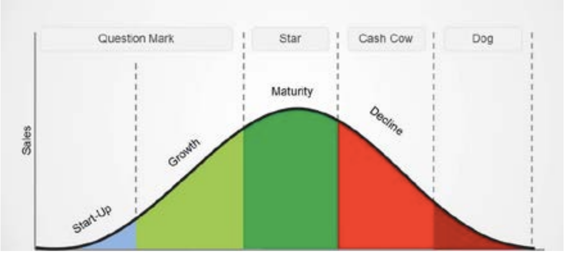


Start up Growth Mature Decline Retire

PayPal Case Study

Shorter interval product lifecycle

- Re-purpose quickly => PIVOT



What is a Palm Pilot?

A Palm Pilot. On display in a museum in Lausanne.

http://www.ecommerce-digest.com/paypal-case-study.html

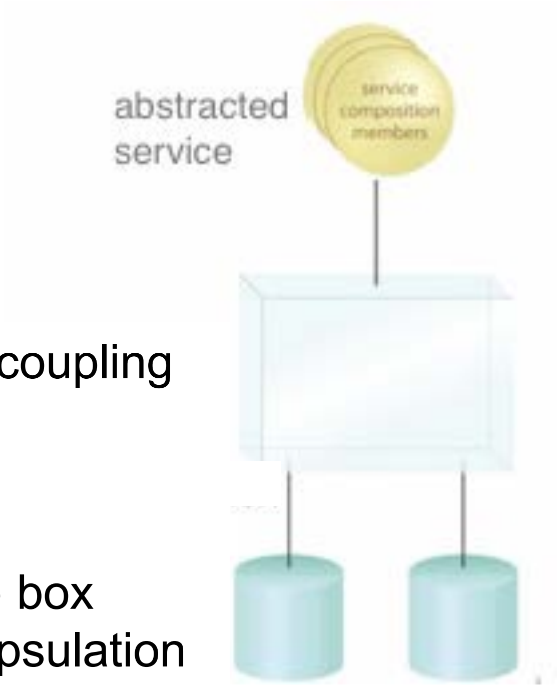
Products that Pivot Quickly

- Flexible services
  - with high internal cohesion
  - with black box encapsulation
  - with abstract interface
    - mix-n-match services into new products
- Interoperable services
  - dissimilar technologies work together dynamically
  - integrate with external 3rd party services
    - PayPal Interoperates with eBay
- Adaptive
  - lightweight development process

Flexible Software Design

- SOA: Service Orientated Architecture
  - mix n match abstract software *services* => *vertical integration*

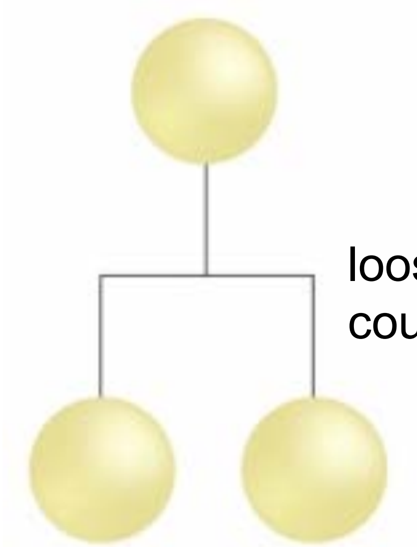
service encapsulating legacy system



tight coupling

white box encapsulation

service encapsulating other services



loose coupling

hidden, black box encapsulation

Why: do Projects?

Are projects still relevant?

- Why not pursue pop-up products?
- Value in quick delivery of products

– Projects have a longer planning horizon

- Allow important but not urgent strategic initiatives
- Allow risk mitigation

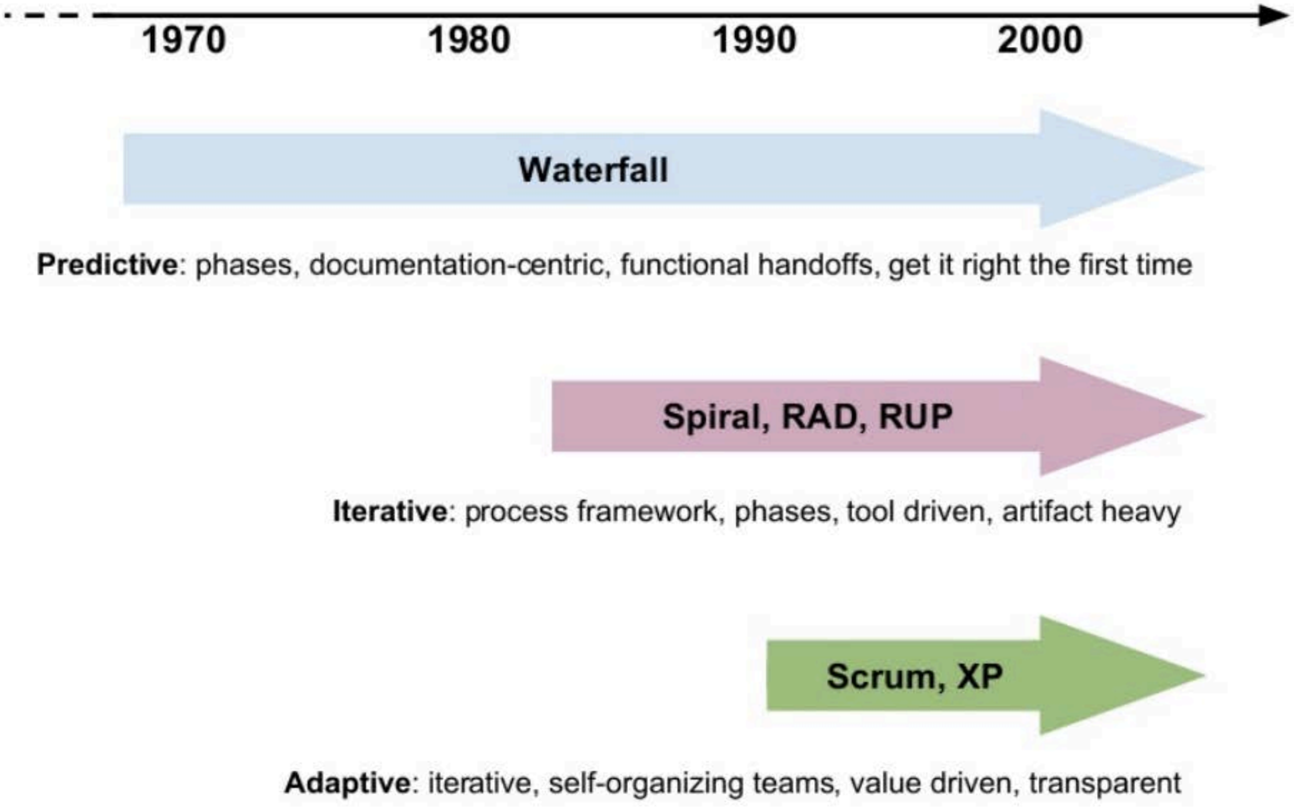
– They provide a team environment

- Collaboration rewards
  – Team outcomes are better than lone wolf developer outcomes

Understanding Software Projects

- History
- Project types
- Success measures
- Complexity measures
- Design measures
- Examples of Large Complex Agile Projects
  - Candy Crush
  - Clash of Clans
  - Frankfurt Airport
- Scrum Team Structure

Project Evolution



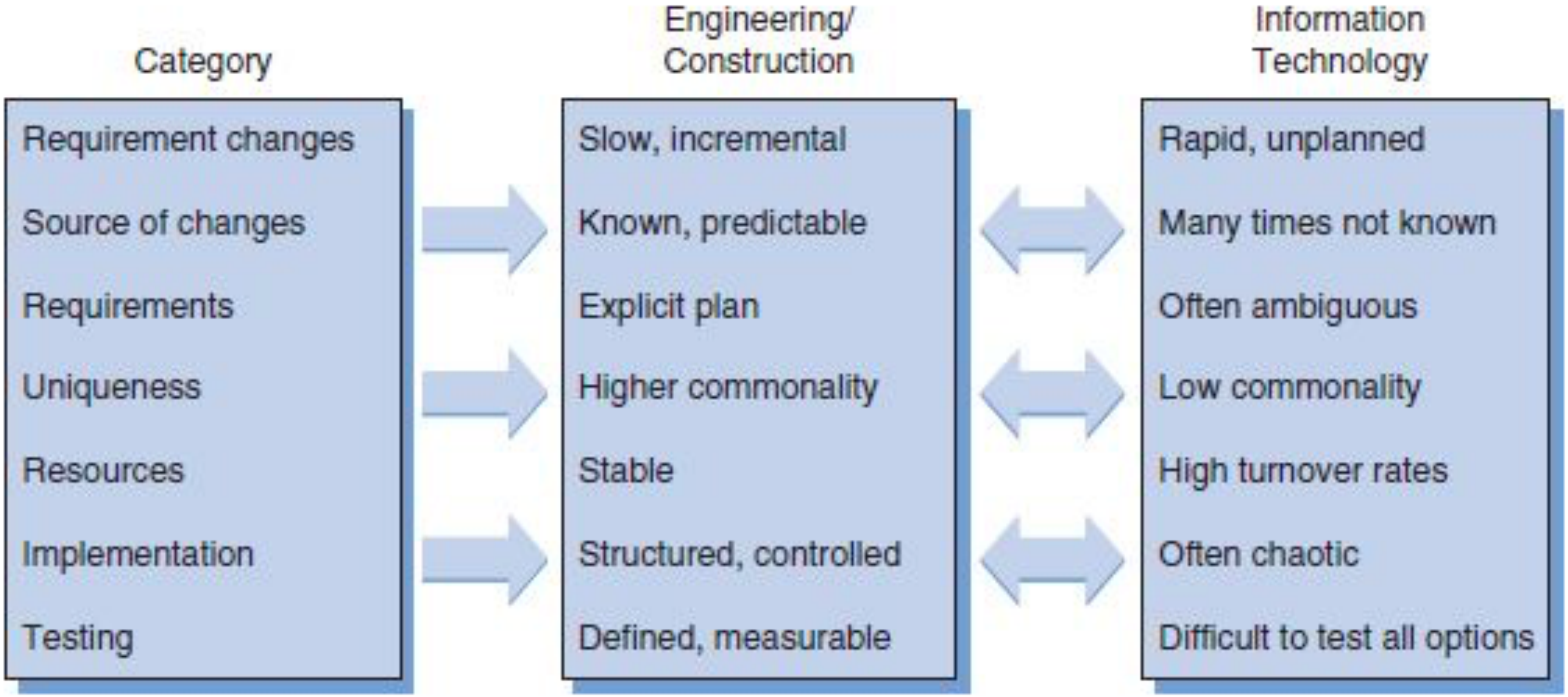
Predictive: phases, documentation-centric, functional handoffs, get it right the first time

Iterative: process framework, phases, tool driven, artifact heavy

Adaptive: iterative, self-organizing teams, value driven, transparent

Project Types

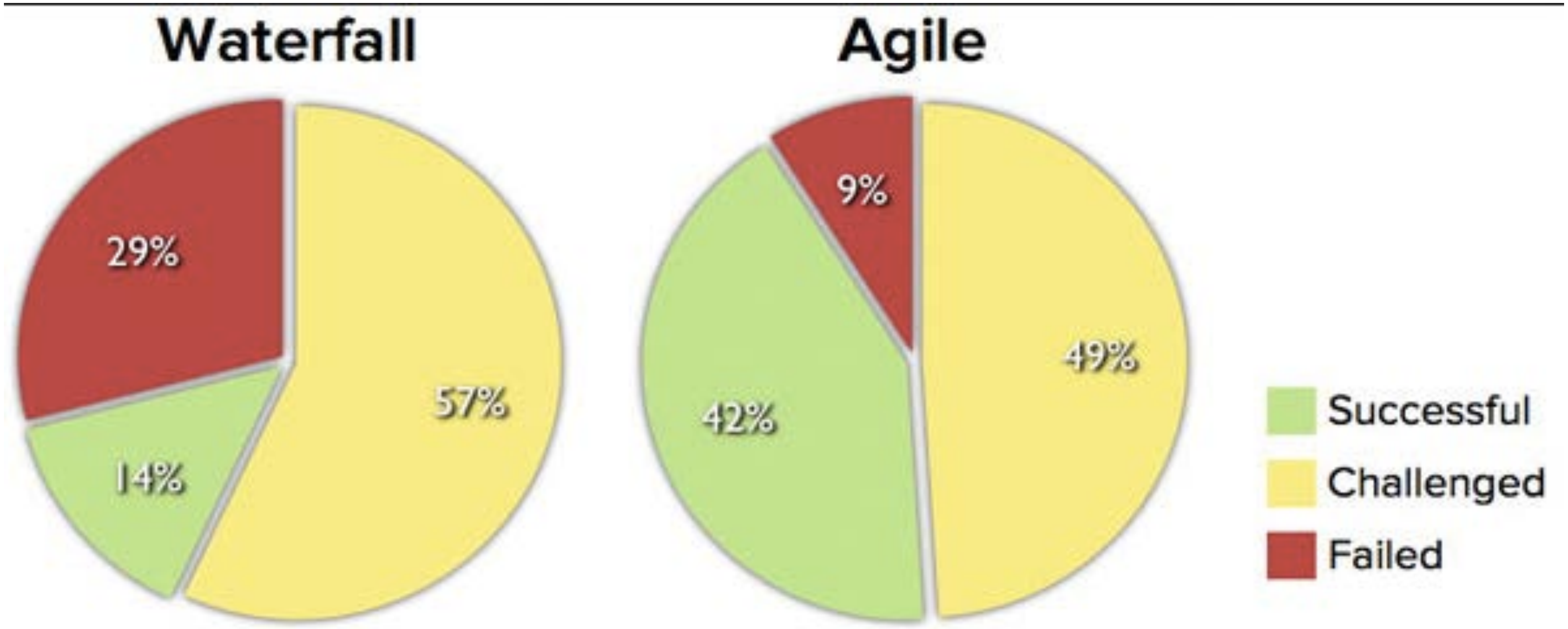
When is Agile-style a good choice?



Category	Engineering/ Construction	Information Technology
Requirement changes	Slow, incremental	Rapid, unplanned
Source of changes	Known, predictable	Many times not known
Requirements	Explicit plan	Often ambiguous
Uniqueness	Higher commonality	Low commonality
Resources	Stable	High turnover rates
Implementation	Structured, controlled	Often chaotic
Testing	Defined, measurable	Difficult to test all options

Which projects perform well?

Think Big, act small



Source: The CHAOS Manifesto, The Standish Group, 2012.

Complexity Measures

Most software projects are complex or complicated.

Universal Problem Solving Technique:

- Break a problem up into smaller problems
- Solve each smaller problem independently

IT Context

- *Loose Coupling* of features
  - reduces the impact of change
- *Dependency Injection* architecture patterns
  - mix n match features

Design definition

Given a IT application with *loose coupling* between components, what does *loose coupling* mean?

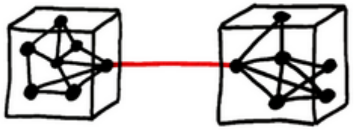
a) there is an insecure, unencrypted channel between them

b) the components have a minimum knowledge of each other

c) the delivery of messages between them is unreliable

d) the destination of a message is not determined

loose coupling

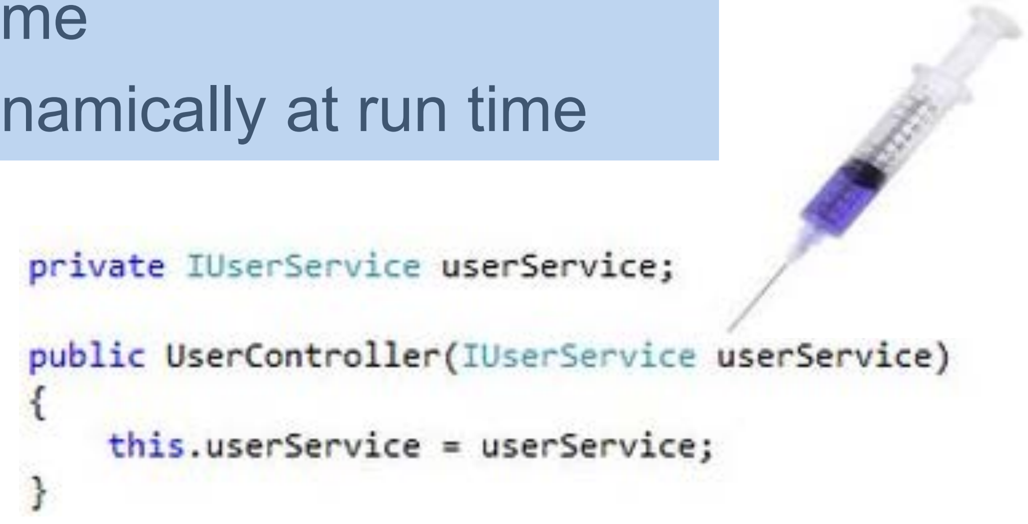


hidden implementation,
black box encapsulation

Design Definition

Given a IT application has the *Dependency Injection* architectural pattern, what can it do with 3rd party code?

a) Use it if it is downloaded locally at compile time
b) Use it if it is downloaded locally at run time
c) Use it remotely at run time
d) Configure and use it dynamically at run time



```
private IUserService userService;

public UserController(IUserService userService)
{
    this.userService = userService;
}
```

https://martinfowler.com/articles/injection.html#InversionOfControl

Large Complex Agile Project

- Example: Candy Crush.
  - Scrum project method
  - 100 million real time online users
  - 70 Big Data analysts
  - 70 Scrum teams
  - 2 week sprints

Large Complex Agile Project

- 2012 Aug - 1st Release
- 2012 Sept, spells
- 2012 Oct, pumpkin bombs
- 2012 Dec, Santa
- 2013 Jan, dark elixir
- 2013 Apr, leagues
- 2013 May, town hall 10
- 2013 Jul, witches
- 2013 Sept, village edit
- 2014, Jan, hero
- 2014 Apr, clan war
- 2014 Sept, lava hound

Clash of Clans
snappy release schedule

Large Complex Agile Project

when you get attacked you get some ideas on what defenses work, what your defensive strategy should be

you get feedback, you tweak the defense to better protect yourself, then someone stronger beats you and you tweak it again

constant loop of: research, come up with a hypothesis, prototype, iterate.

every loss is not a mistake, but an opportunity to learn and get better.

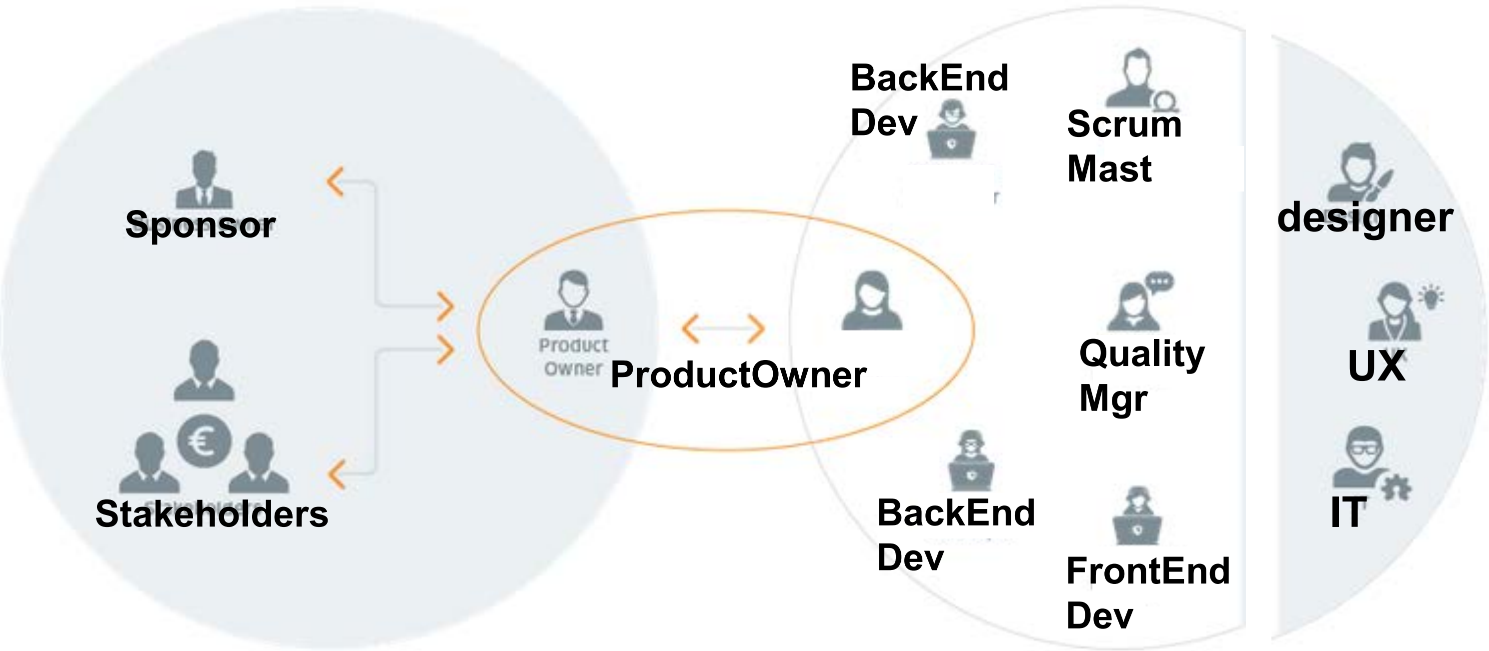
Clash of Clans game play: a metaphor for Scrum

Large Complex Agile Project

- Example: Frankfurt Airport Omni-channel, 2016.
  - enterprise web application
  - 10th busiest airport, biggest RETAIL airport in the world
  - digital retail e-commerce product
  - developer: AOE
    - aoe.com
  - **Developed using Scrum**
  - presumes changes will occur and plans are conditional

Scrum Development Team

- **Dev Team:**
  - usual team: backend (BE), frontend (FE) , quality assurance (QA)
  - specialists team: designer, UX, software architect, IT



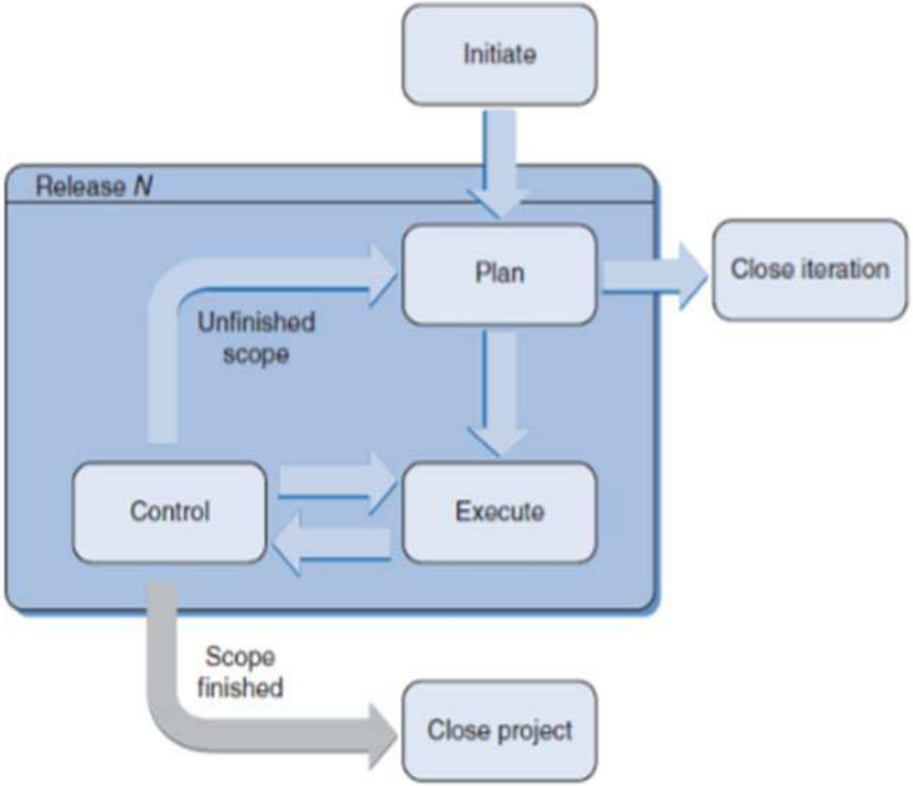
https://www.aoe.com/en/company/agile-teams.html

Common PMP Frame

- Formal Stages
- Agile Stages
- Common Initiation Stage
- Different Scope Technique

Formal Stages

Formal PMP lifecycle



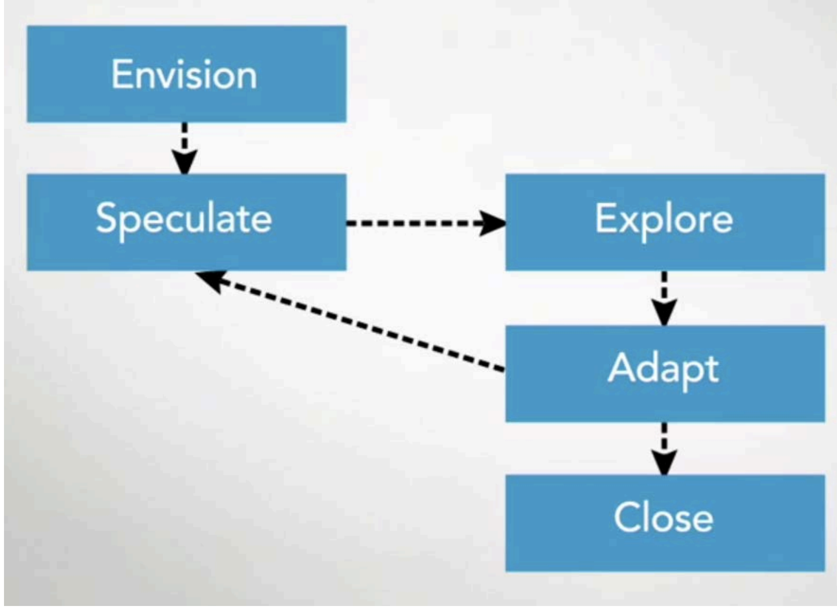
What is the goal?

Deliver a product, eventually.

Agile Stages

Agile PMP lifecycle

Looks pretty similar ☺



What is the goal?

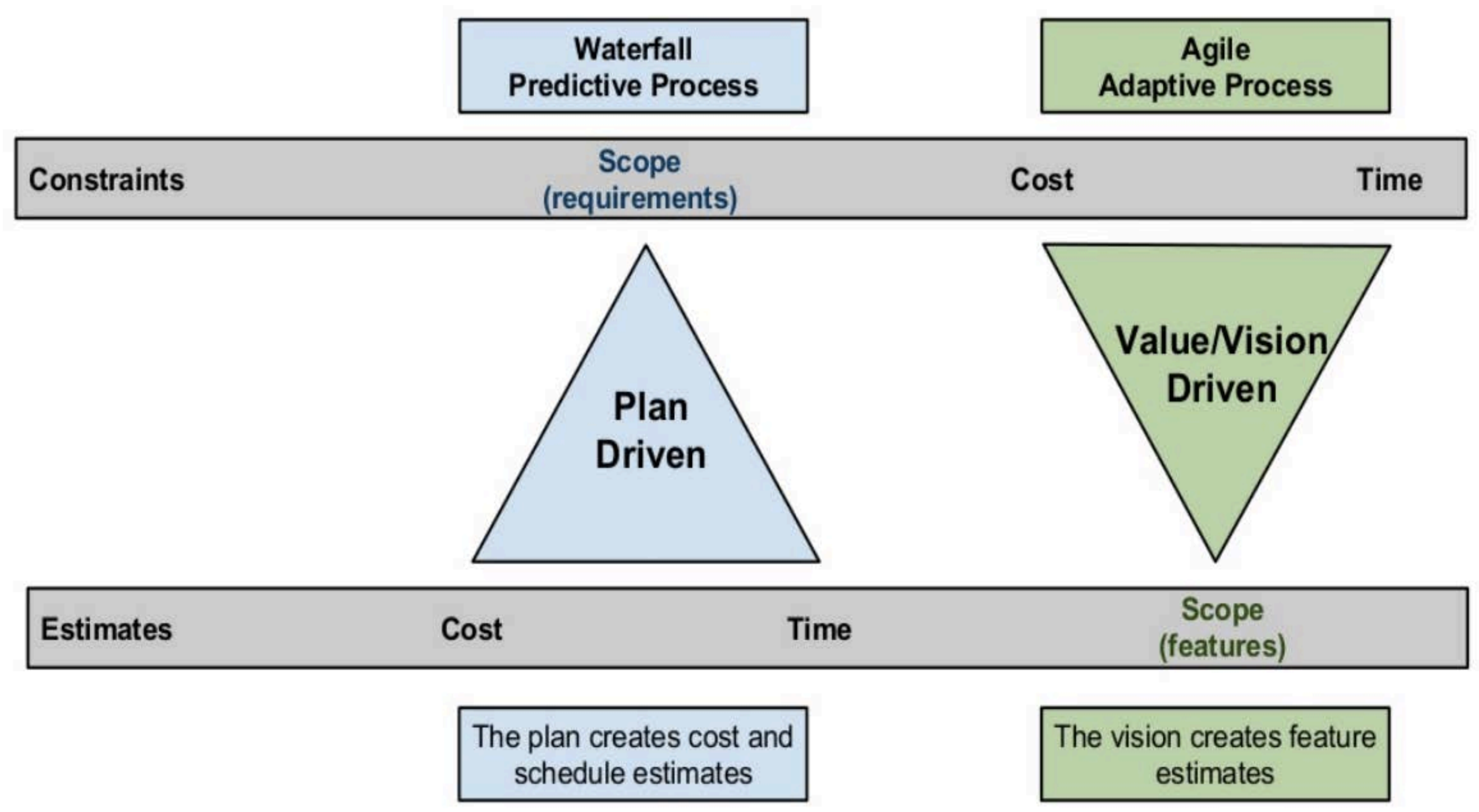
Deliver frequent small chunks of product.

Common Characteristics

All Projects have a Project Charter

- – The Inception Stage
- Recognize the end goal
- Protocol to communicate
- Stakeholder management
- Formal approval
- Follow an accepted project lifecycle
  - – Delivery date
- Demonstrate understanding of requirements
- Build a dedicated and focused team
- Share and manage a schedule

Different Scope Perspective



Waterfall
Predictive Process

Agile
Adaptive Process

Constraints

Scope
(requirements)

Cost

Time

Plan
Driven

Value/Vision
Driven

Estimates

Cost

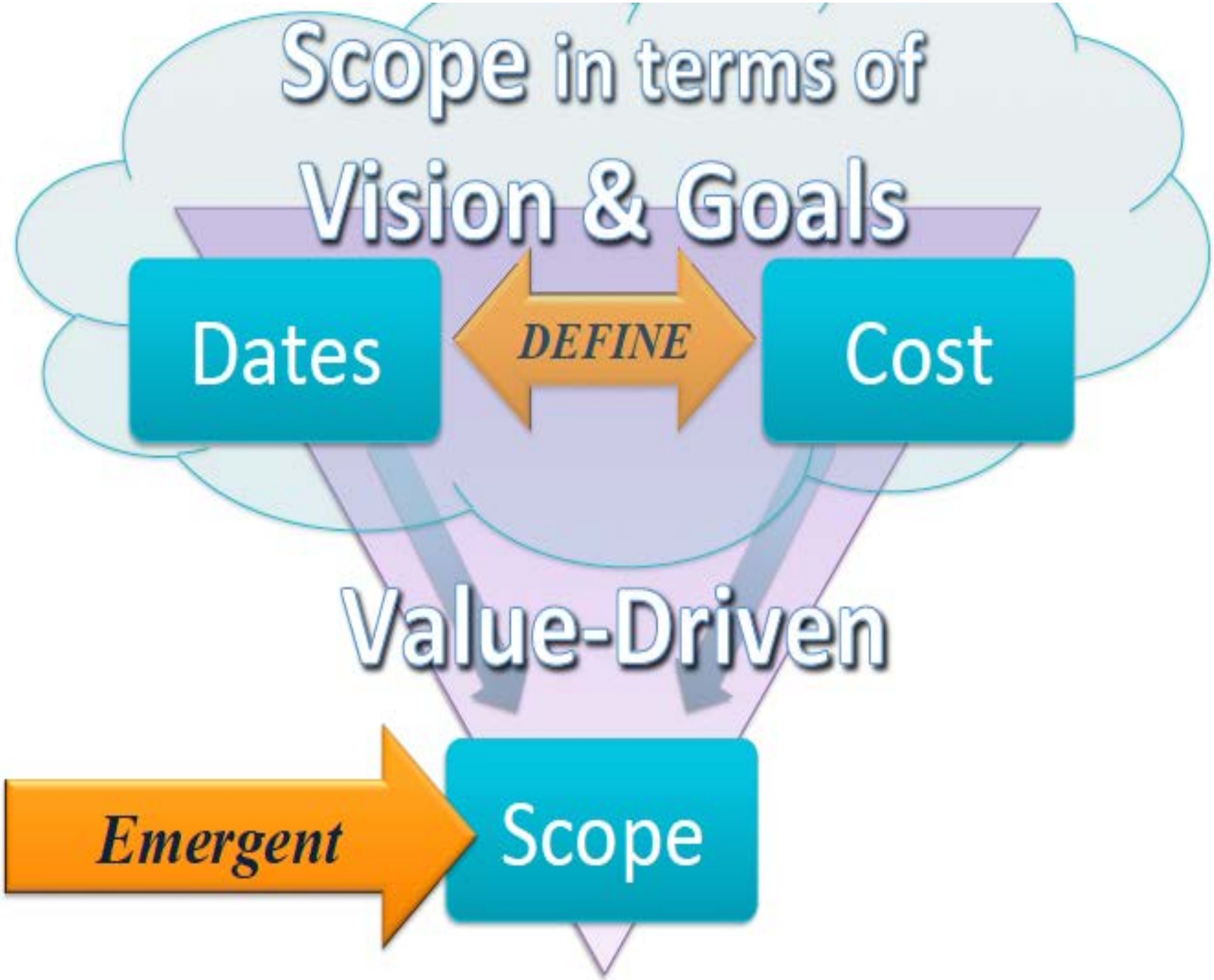
Time

Scope
(features)

The plan creates cost and schedule estimates

The vision creates feature estimates

Agile Scope


Scope in terms of
Vision & Goals
Dates
DEFINE
Cost
Value-Driven
Emergent
Scope

Agile PMP Stages

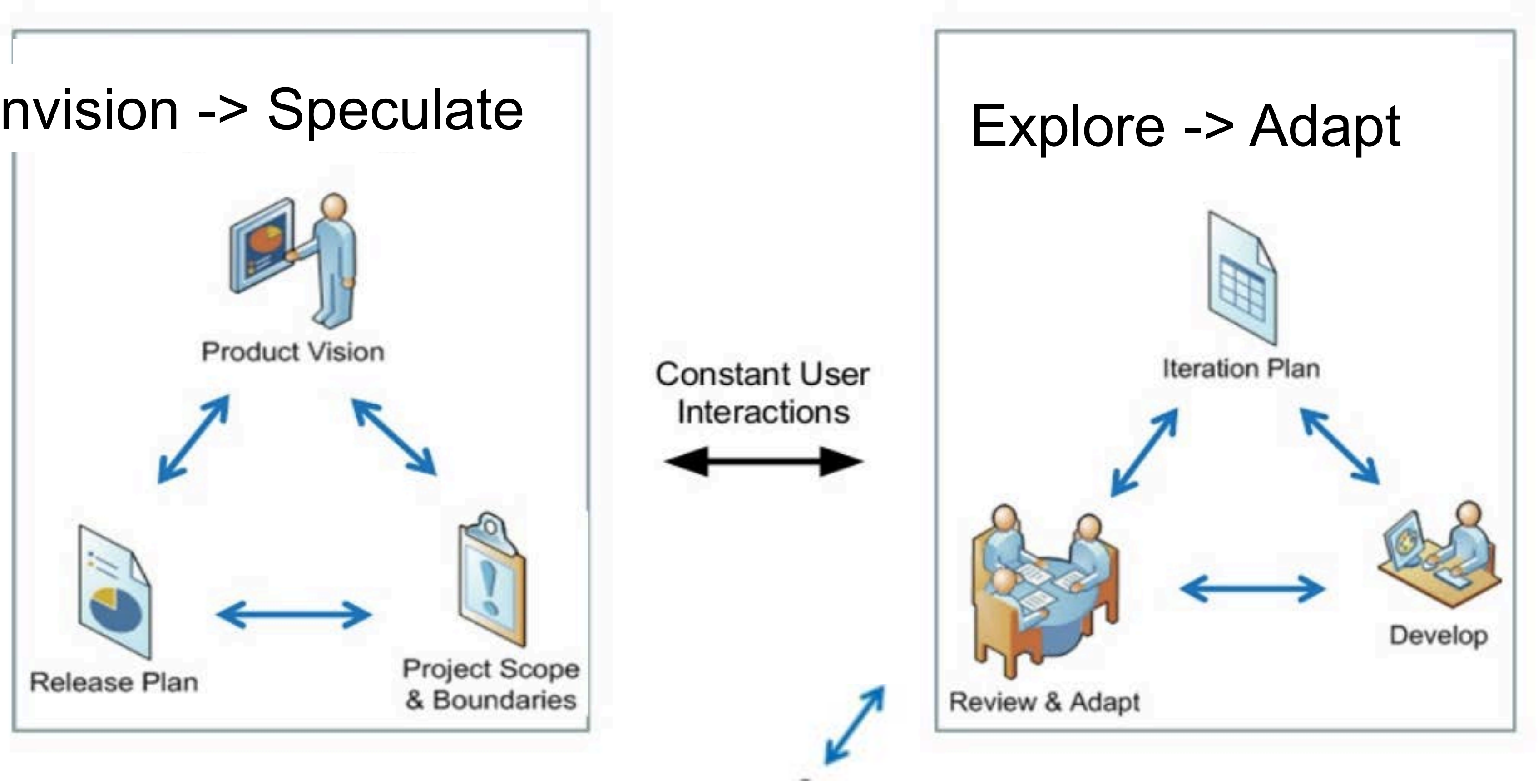
- Envisage
- Speculate
  - High level features
- Explore
  - End on time
  - Project Managers role
- Adapt
- Close

Agile PMP Overview

Agile PMP stages map to original SW Development process

Envision -> Speculate

Explore -> Adapt



Close

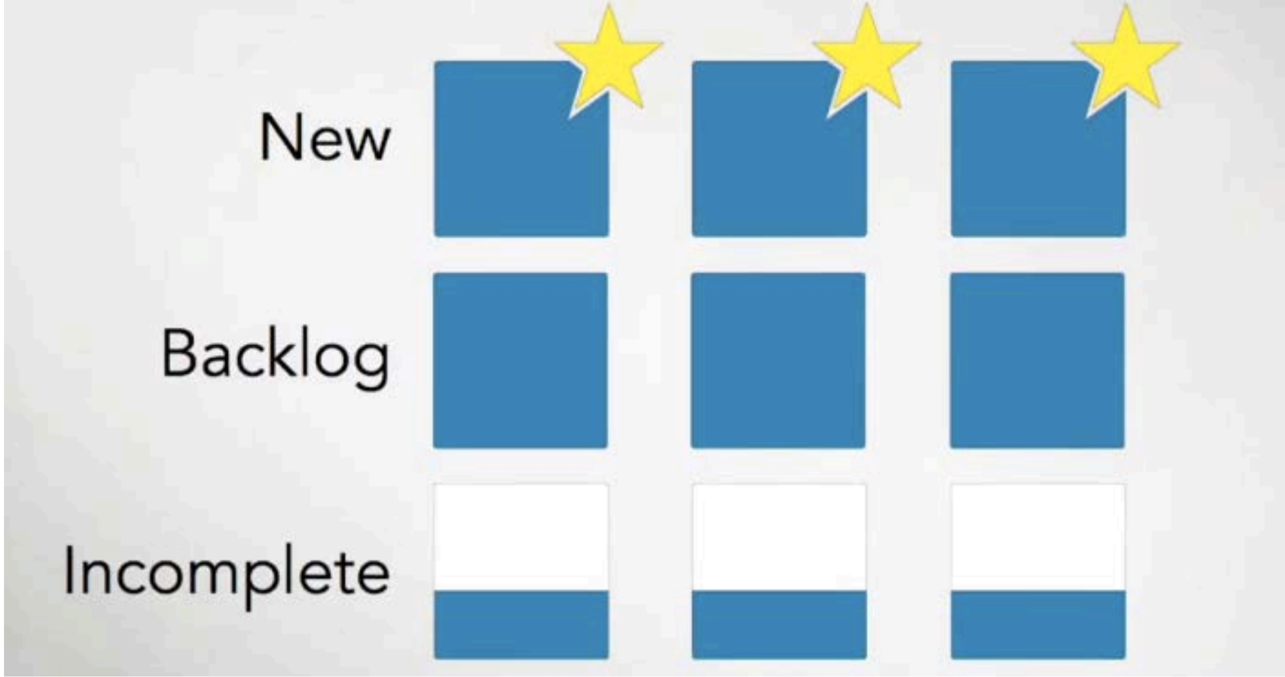
Agile Project: Envision Stage

- Project Charter
- Project Tool Set

- Project Risk Register

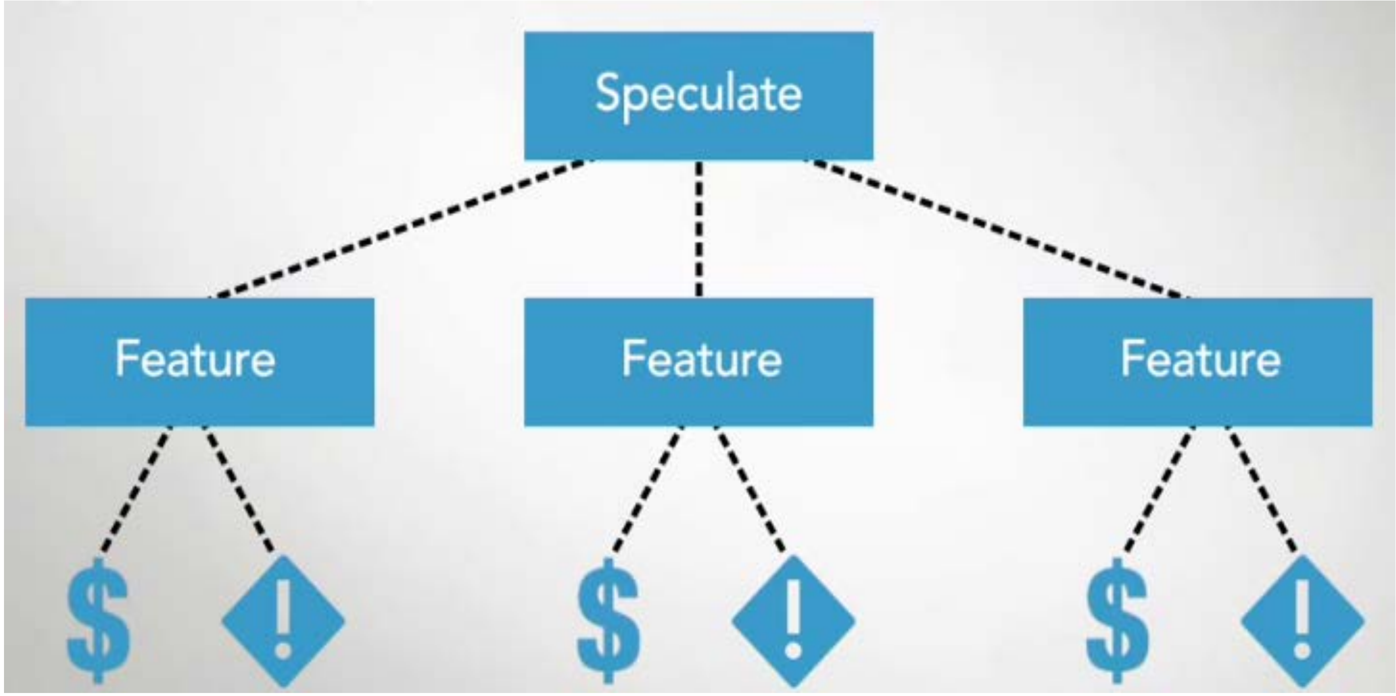
Agile Project: Speculate Stage

- The Backlog of features



Agile Project: Speculate Stage

- Placeholder for conversations
- Organize Priorities



Plan for next stage

- Create:
  - Iteration plan
  - Milestones
  - Release plan

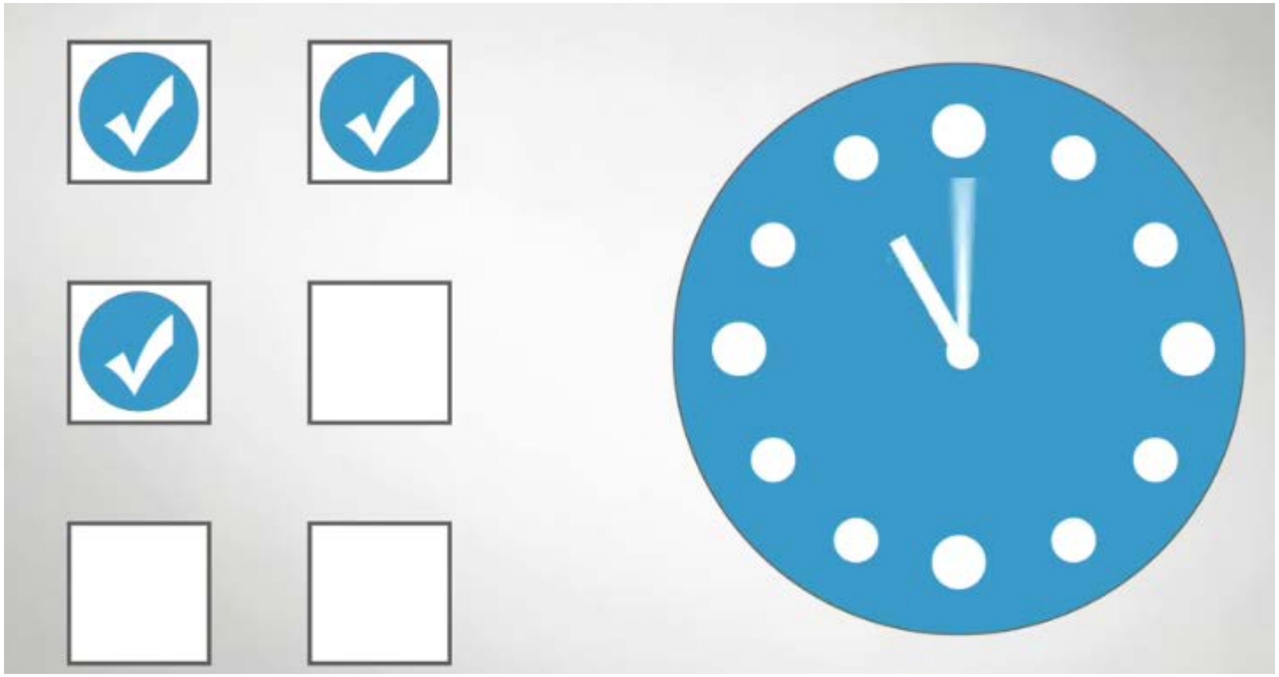
Agile Project: Explore Stage

The Sprint!

- Conversations and collaborations
- Explore => code stuff !

Agile Project: Explore Stage

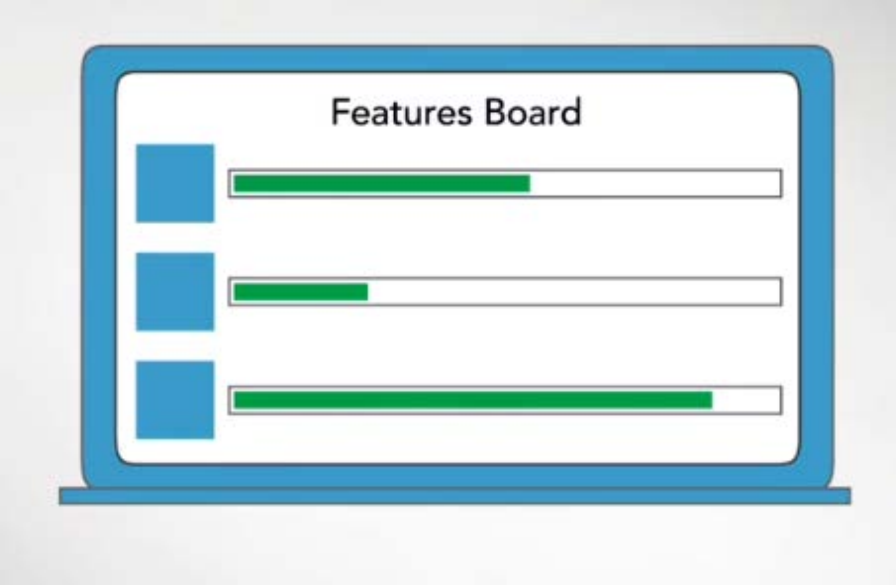
- End Sprint on schedule, not when all features done
- Need to establish the team's Velocity



Agile Project: Explore Stage

- Project manger’s role is as an observer

Scrum Master



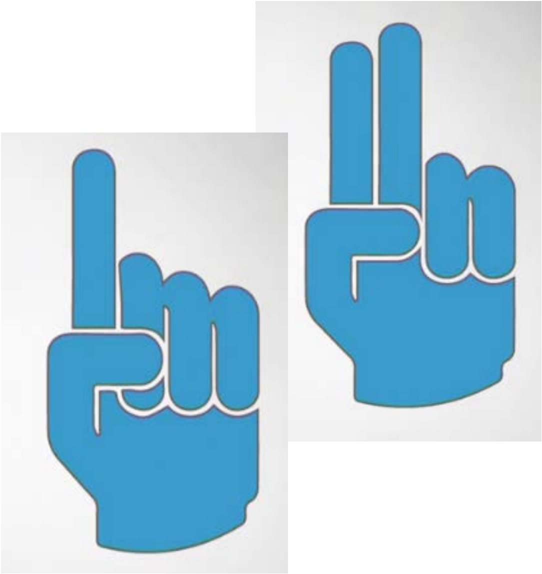
  - Self organizing teams
  - Visual progress on display
  - Everyone knows the status
  - Nowhere to hide

Agile Project: Adapt Stage

- Be open and critical: avoid ceremony
- Brainstorm important issues
- Everyone has a voice
- Collect multiple alternative solutions to problems
- Vote on which solution will be adopted

Fist of Five

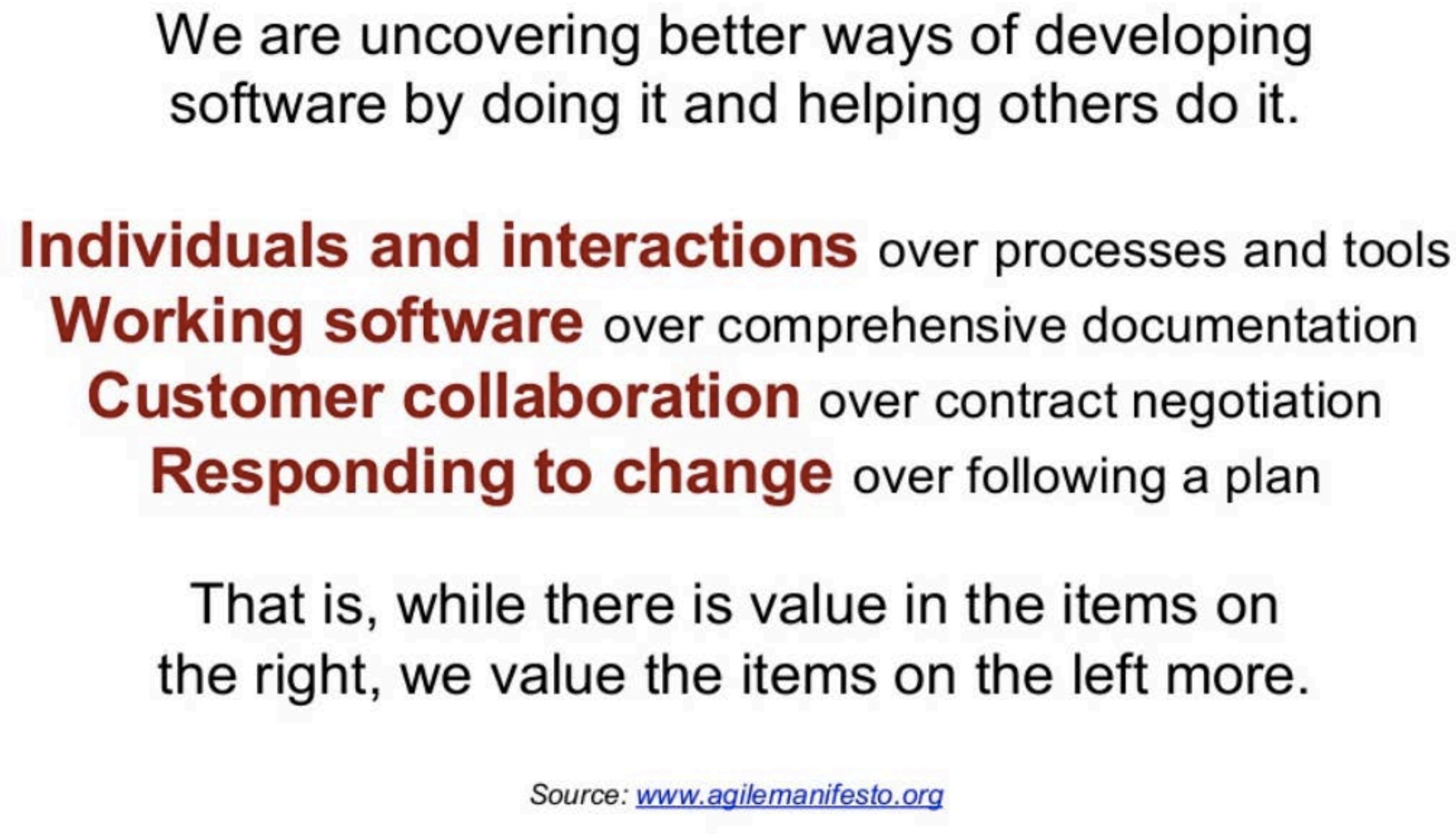
5 – love it!
4 – happy with it
3 – can accept
2 – reservations
1 – grave misgivings



Agile SW DEV: Overview

- Revision
    - Agile Manifesto
    - 12 Principles
- The Agile Tribe
- Popularity of each tribe
- Common tribal behavior
- Different behavior between tribes
    - Plus/Minus/Interesting
- Example of visual boards

Manifesto



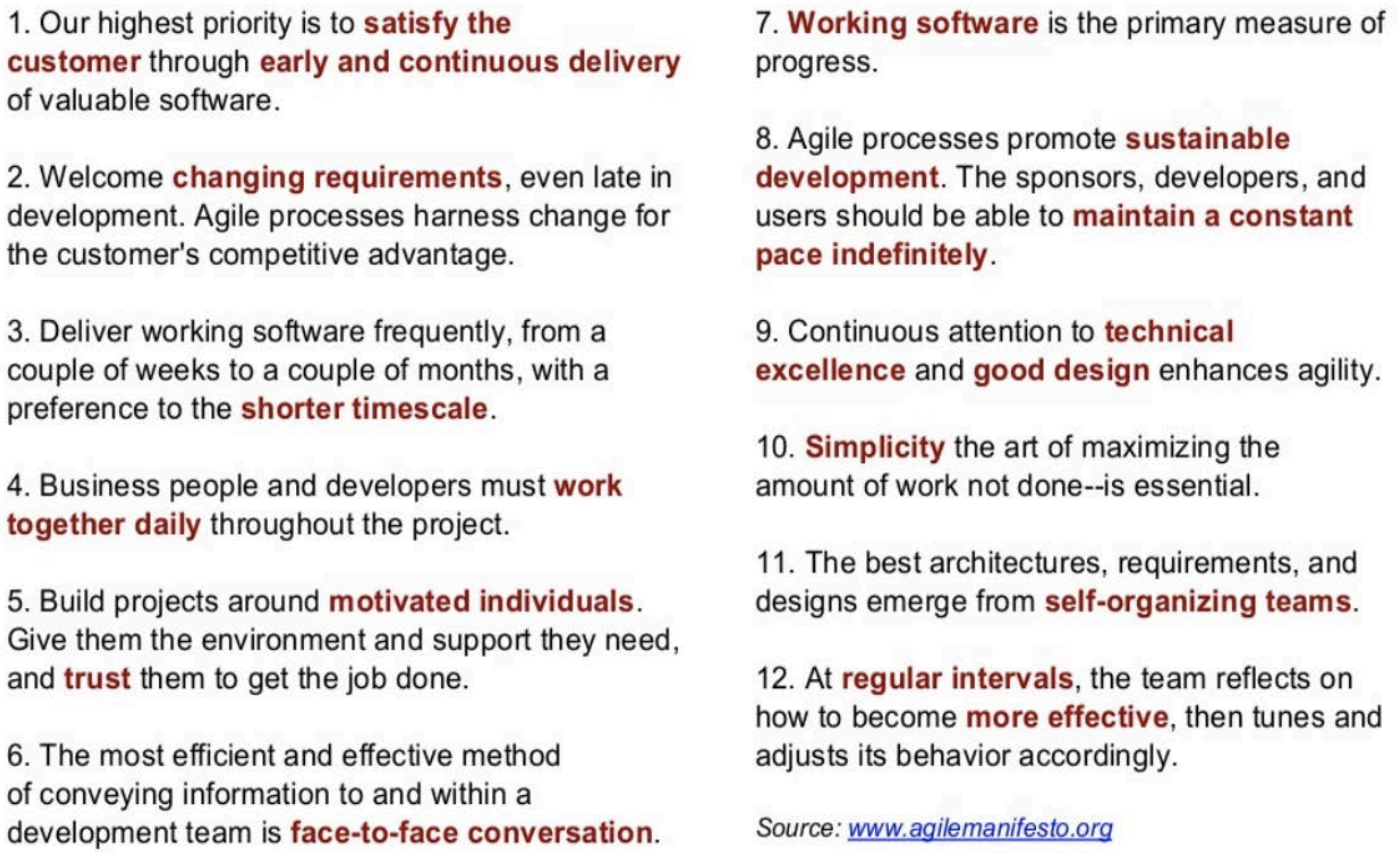
We are uncovering better ways of developing software by doing it and helping others do it.

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Source: www.agilemanifesto.org

The 12 Principles



1. Our highest priority is to **satisfy the customer** through **early and continuous delivery** of valuable software.

2. Welcome **changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the **shorter timescale**.

4. Business people and developers must **work together daily** throughout the project.

5. Build projects around **motivated individuals**. Give them the environment and support they need, and **trust** them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation**.

7. **Working software** is the primary measure of progress.

8. Agile processes promote **sustainable development**. The sponsors, developers, and users should be able to **maintain a constant pace indefinitely**.

9. Continuous attention to **technical excellence** and **good design** enhances agility.

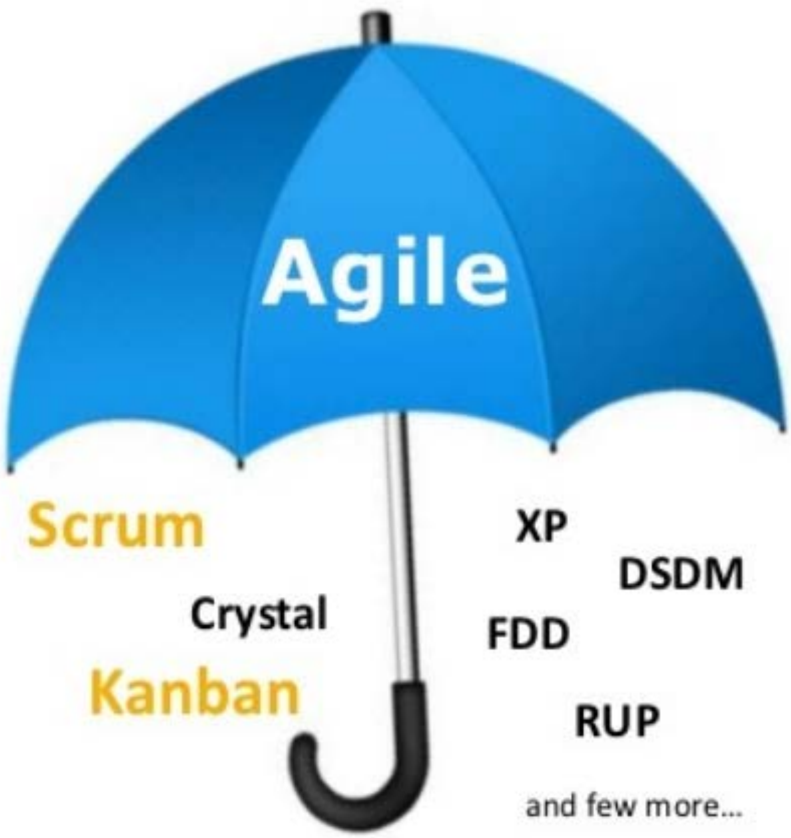
10. **Simplicity** the art of maximizing the amount of work not done--is essential.

11. The best architectures, requirements, and designs emerge from **self-organizing teams**.

12. At **regular intervals**, the team reflects on how to become **more effective**, then tunes and adjusts its behavior accordingly.

Source: www.agilemanifesto.org

The Tribes



Agile

Scrum

Crystal

Kanban

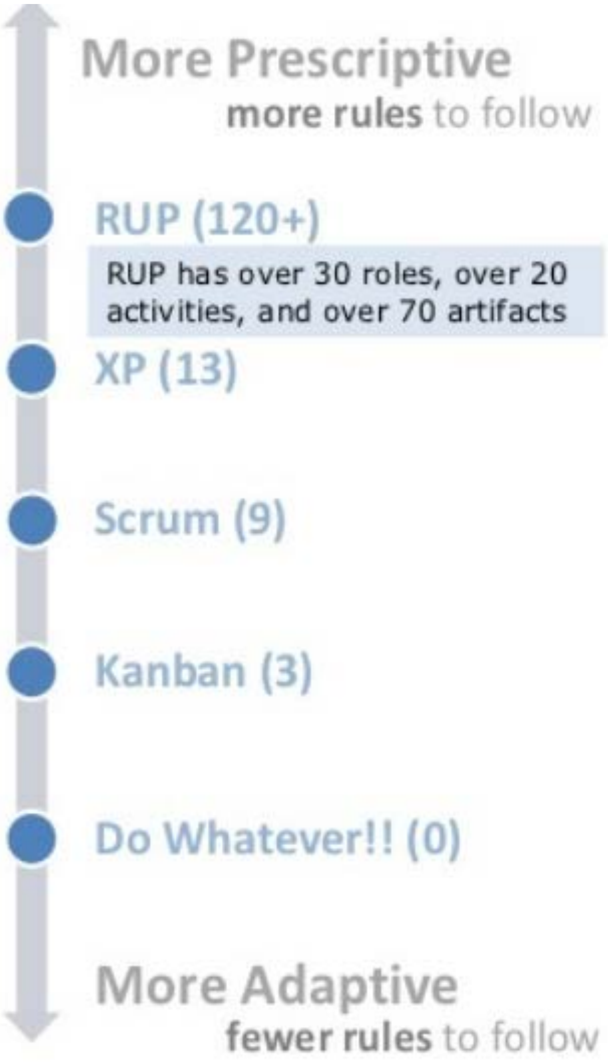
XP

DSDM

FDD

RUP

and few more...



More Prescriptive
more rules to follow

RUP (120+)

RUP has over 30 roles, over 20 activities, and over 70 artifacts

XP (13)

Scrum (9)

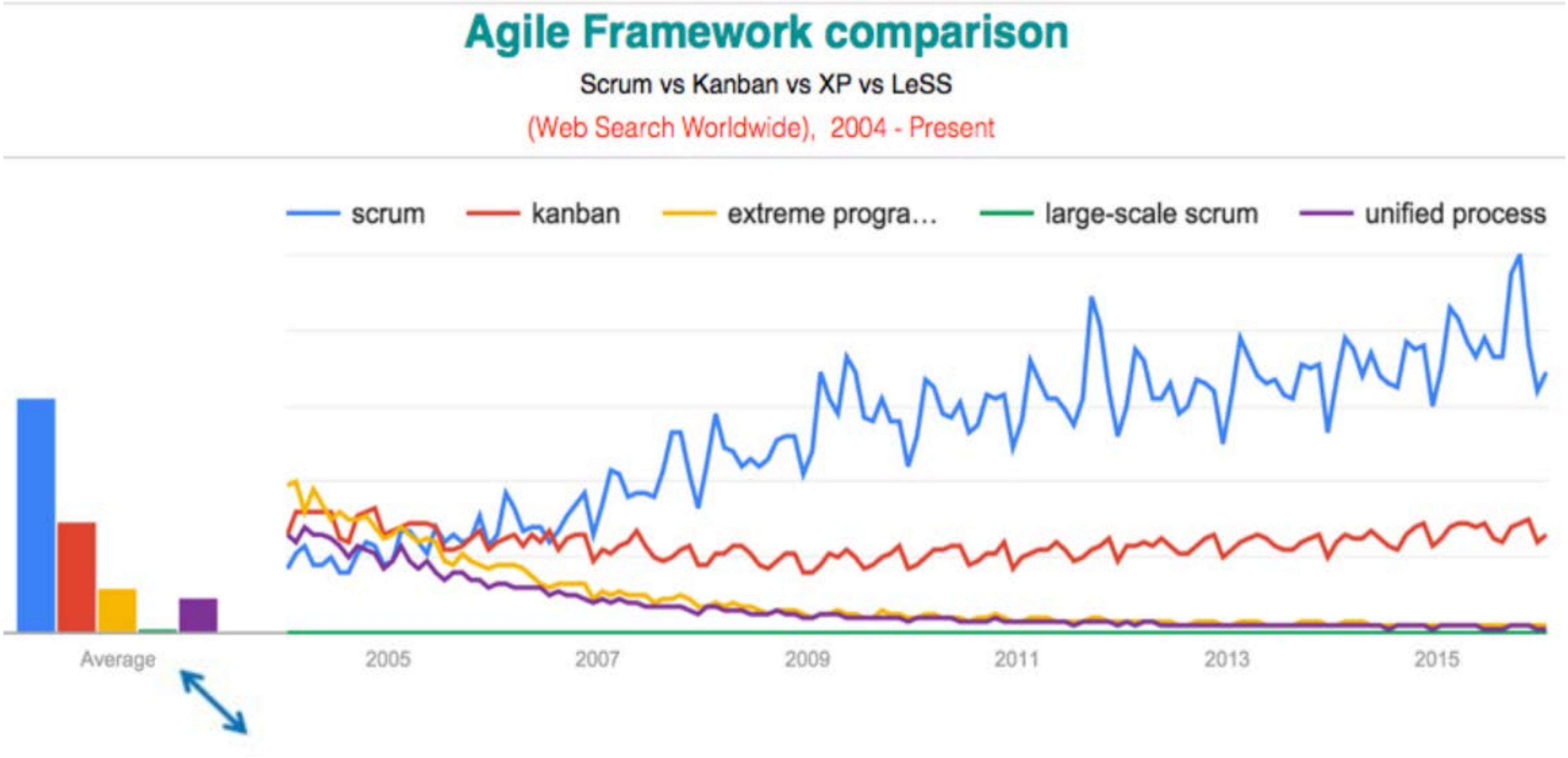
Kanban (3)

Do Whatever!! (0)

More Adaptive
fewer rules to follow

* Check wikipedia for list of all Agile methods

Tribal Popularity



Common Culture

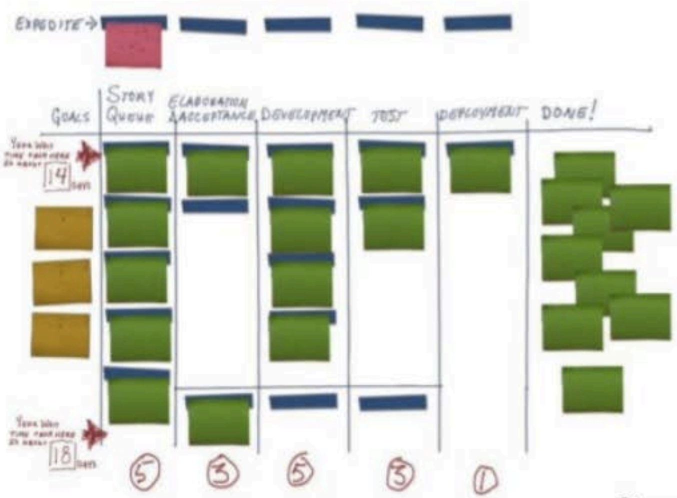
+ Visual Task Board
+ Burn Down chart
+ Burn Up chart
+ Test Driven Development

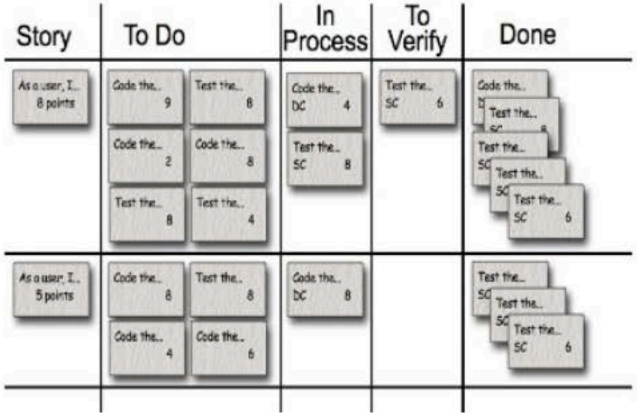
- Design Phase can be less rigorous
- Refactoring time slot can be overtaken by new initiatives

Tribal Plus/Minus/Interesting

- Scrum – has Sprints, self-organizing team
  - + Deliver chunks of stable and packaged code
  - + Time boxed Sprint fosters design opportunity
  - ? Meetings surround each Sprint can become ceremonial
- Extreme Programming – prescriptive
  - + Deliver chunks of simple code quickly
  - - ”TODO” list of features, without design structure
  - ? Pair programming
- Kanban – intuitive, visual, smooth work flow
  - + Widely adopted
  - + Deliver code with “Just-In-Time” efficiently
  - - Production line of features, without design frame

Different Visual Boards

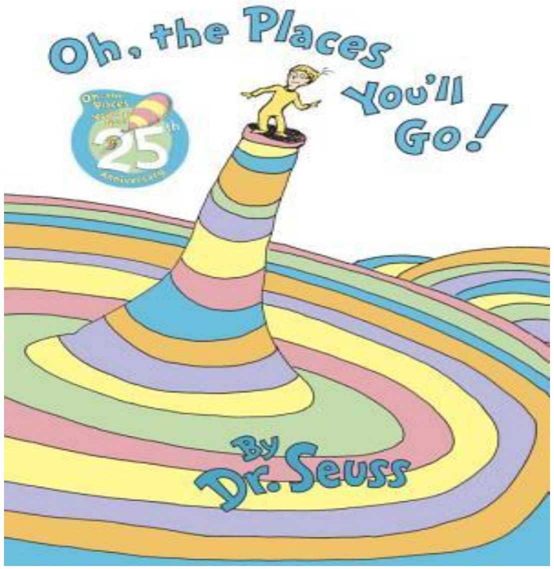




Story	To Do	In Process	To Verify	Done
As a user, I... 8 points	Code the... 9 Test the... 8 Code the... 2 Code the... 8 Test the... 8 Test the... 4	Code the... DC 4 Test the... SC 8	Test the... SC 6	Code the... Test the... SC Test the... SC Test the... SC Test the... SC 6
As a user, I... 5 points	Code the... 8 Test the... 8 Code the... 4 Code the... 6	Code the... DC 8		Test the... SC Test the... SC Test the... SC 6

References

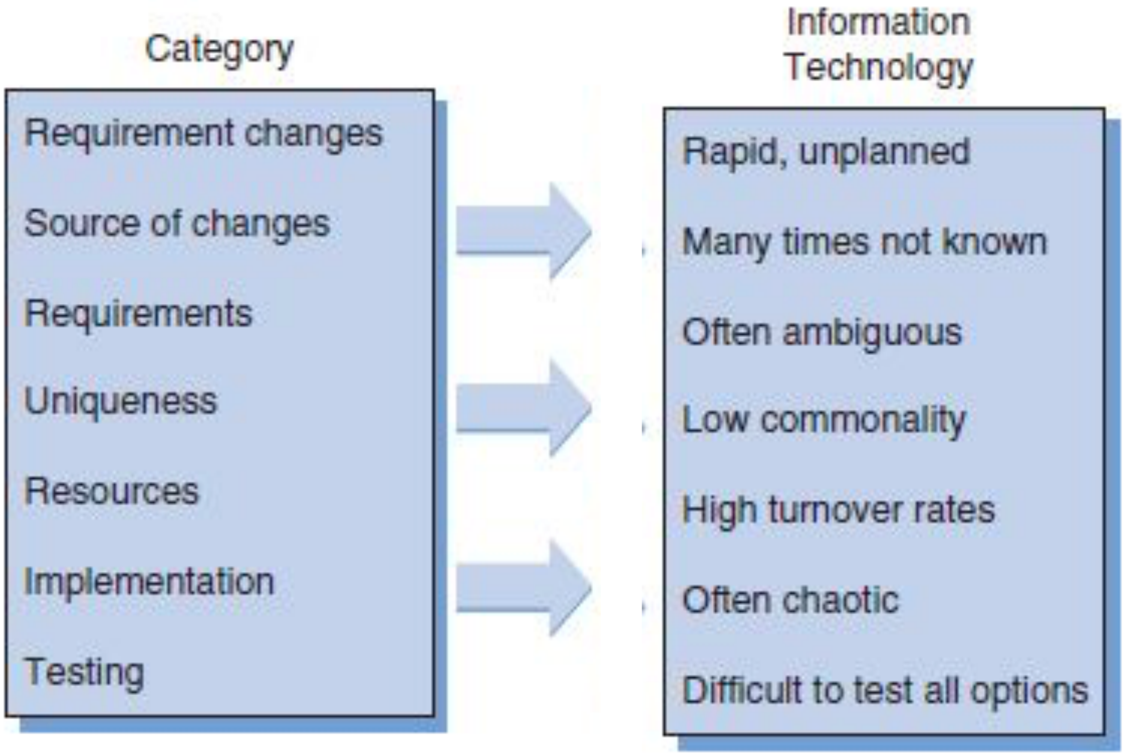
- Agile Software Development with Scrum. Ken Schwaber and Mike Beedle
- Chaos Manifesto https://larlet.fr/static/david/stream/ChaosManifesto2013.pdf
- Kanban: Successful Evolutionary Change for Your Tech Business. David J. Anderson
- Kanban https://www.atlassian.com/agile/kanban
- Big Data
  - http://www.irlogi.ie/wp-content/uploads/2013/10/Big-Data-Analytics-and-GI-KieranTowey.pdf
- Pivot Case Studies
  - https://blog.continuum.net/5-cases-of-companies-that-successfully-pivoted-business-models
- PayPal Case Study
  - http://www.ecommerce-digest.com/paypal-case-study.html
- Are Projects Dead?
  - https://www.mountaingoatsoftware.com/blog/is-it-time-to-stop-thinking-about-projects
- AOE https://www.aoe.com/en/company/agile-teams.html
- Torak www.torak.com
- The New New Product Development Game.
  - Takeuchi, Hirotake, Nonaka, Ikujiro.
- Dr Seuss



Purpose of lecture

1. What is trending in IT, the knowledge industry?
   - Big Data
2. What is happening to products?
   - Quick Pivots
3. Why do products?
   - Strategic Initiatives, Team Building
4. When is an Agile project a good choice?

When to choose Agile



Category	Information Technology
Requirement changes	Rapid, unplanned
Source of changes	Many times not known
Requirements	Often ambiguous
Uniqueness	Low commonality
Resources	High turnover rates
Implementation	Often chaotic
Testing	Difficult to test all options

Why: New Opportunities

Next Lecture

- How to participate in an Agile process
- How to do the Group Scrum Assignment
- How to do Burn Down Charts

Any questions?